APPENDIX VIII - REBOOT EVENT

```
import java.lang.*;
public class RebootEvent implements IMessageEvent
{
  public final static int REBOOT_BUTTON_PRESSED = 0x30000001;
  public final static int REBOOT_BY_IMPLEMENTATION = 0x30000002;
  public final static int REBOOT_BY_TRUSTED_APP = 0x30000003;
  public final static int REBOOT_FOR_UNHANDLED_SYS_ERROR = 0x30000004;
  public final static int REBOOT_FOR_UNHANDLED_APP_ERROR = 0x30000005;

  // Static reboot event Id number.  Incremented for each new event.
  private static long lastEventId;

  private long eventId;
  private long appId;
  private int typeCode;
  private long date;
  private String message;

  public RebootEvent(int typeCodeIn, long appIdIn, String messageIn)
  {
    lastEventId = (lastEventId == Long.MAX_VALUE) ? 0 : lastEventId + 1;
    eventId = lastEventId;
    typeCode = typeCodeIn;
    appId = appIdIn;
    date = System.currentTimeMillis();
    message = messageIn;
  }

  /**
   * Get the unique identifier of the event.
   *
   * @return unique identifier of the event.
   */
  public long getEventId()
  {
    return eventId;
  }

  /**
   * Get the unique identifier of the application logging the event.
   *
   * @return The application identifier.
   */
  public long getAppId()
  {
    return eventId;
  }

  /**
   * Get the event type code.  Identifies a code specific to the event system.
   *
   * @return type code of the event.
   */
  public int getTypeCode()
{
  return typeCode;
}

/**
 * Set the error type code.  Allows an application to change the error received.
 *
 * @param error - The new error value.
 */
public void setTypeCode(int typeCodeIn)
{
  typeCode = typeCodeIn;
}

/**
 * Get the event date in milli-seconds from midnight January 1, 1970.
 *
 * @return Date the event was submitted to the system.
 */
public long getDate()
{
  return date;
}

/**
 * Set the date.  Allows an application to change the date received.
 *
 * @param error - The new error value.
 */
public void setDate(long dateIn)
{
  date = dateIn;
}

/**
 * Get the readable message.
 *
 * @return the readable event message.
 */
public String getMessage()
{
  return message;
}
protected void finalize()
{
  // Store the lastRebootEventId in a persistent storage object.
}
static {
  if(/*some persistent storage object exists with reboot event Id stored in it*/ true)
    lastEventId = /* value in the persistent storage object */1;
  else
     lastEventId = 0;
}

                                              }
```